LAB 3

STAT 28

February 02, 2017

Welcome to the lab 3! In this lab, you will

1) plot your density curves and violin plot;
2) obtain some basic statistics by groups;
3) implement a permutation test;

We will continue to use the rent price dataset from the lab 2. In the table **craigslist.csv**, each posting record (row) contains the following information:

- time: posting time
- price: apartment/housing monthly rent price
- size: apartment/housing size (ft^2)
- brs: number of bedrooms
- title: posting title
- link: posting link, add "https://sfbay.craigslist.org" to visit the posting page
- location: cities

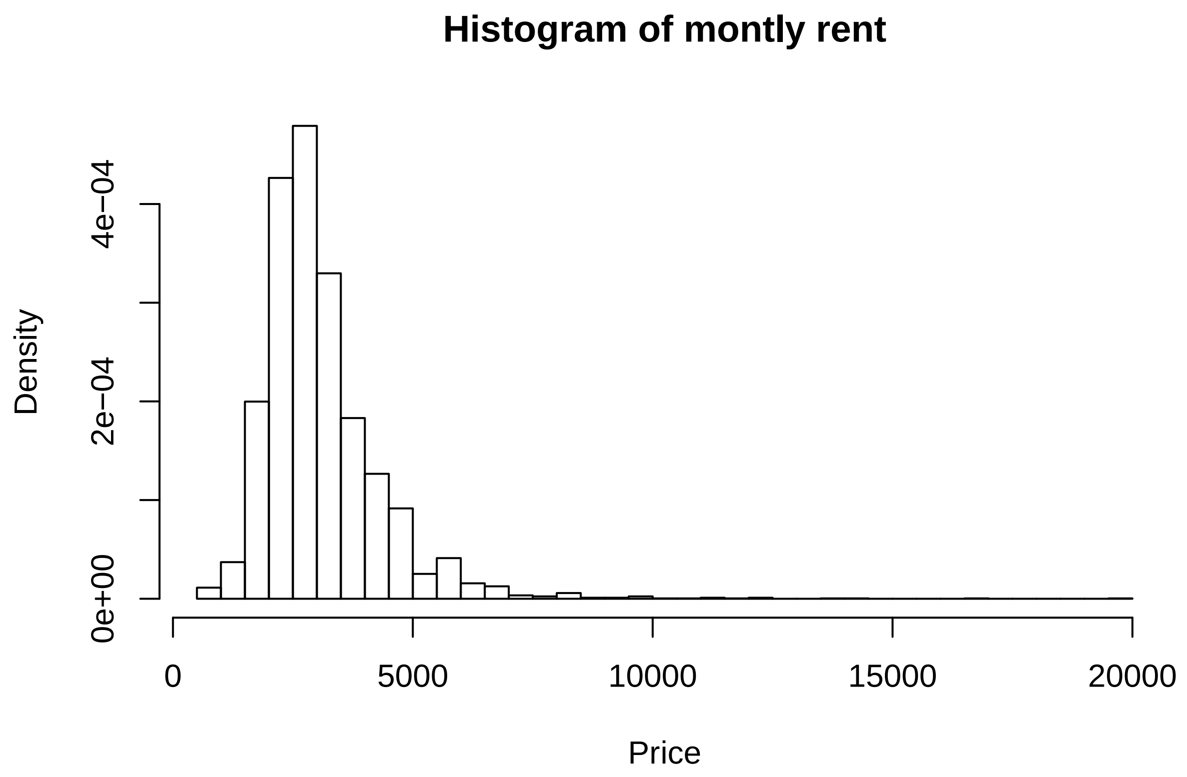
Read in data. Create `one.bedrooms` data frame.

```
craigslist <- read.csv("craigslist.csv",
                       header = TRUE)
one.bedrooms <- craigslist[craigslist$brs == 1,]
```

Density Curves and Violin Plot

Recall the histogram you plotted with `freq = FALSE` in lab 2:

```
hist(craigslist$price, xlab = "Price", freq = FALSE, breaks = 50,
     main = "Histogram of montly rent")
```



Exercise 1

Now fit a density curve for the monthly rent histogram. Compare the y axis with the histograms you draw. (HINT: Use the function `density` and refer to *Example of Flight data* in section 4.3.1 (Kernel density estimation) of Lecture 1.)

```
# insert your code here
```

Exercise 2

Draw the violin plot of one bedroom rent price by cities. (Follow the example given in lectures (section 4.4.3), do not forget to add titles and legend)

```
library(vioplot)
```

```
## Loading required package: sm
```

```
## Package 'sm', version 2.2-5.4: type help(sm) for summary information
```

```
source("http://www.stat.berkeley.edu/~epurdom/RcodeForClasses/myvioplot.R")
# insert your code here
```

Summarize dataset by groups

In this dataset, we are more interested in the summaries of rent price by cities. The `tapply` function is useful when we need to break a vector into groups (subvectors), and apply a function (for example, the `mean` function) within each group. The usage is:

```
tapply(Variable_of_interests, Factor_vector_representing_groups, Function_you_want_to_apply_on)
```

For example, to obtain the median rent price by cities.

```r
tapply(craigslist$price, craigslist$location, median)
```

```
##             alameda albany / el cerrito            berkeley 
##              2497.5              2300.0              2885.0 
##          emeryville          menlo park       mountain view 
##              3095.0              3600.0              2862.5 
##             oakland           palo alto        redwood city 
##              2495.0              3395.0              3152.0 
##            richmond           sunnyvale 
##              2850.0              2625.0 
```

You can write and apply your own functions. For example, to get the percentage of rent price less than \$2000/month by city.

```r
tapply(craigslist$price, craigslist$location, function(x){mean(x < 2000)})
```

```
##             alameda albany / el cerrito            berkeley 
##          0.16500000          0.21621622          0.10101010 
##          emeryville          menlo park       mountain view 
##          0.03809524          0.01285347          0.08555133 
##             oakland           palo alto        redwood city 
##          0.27809308          0.03802817          0.06333973 
##            richmond           sunnyvale 
##          0.18625277          0.08795812 
```

The rent price in Berkeley is much better than Palo Alto! The median monthly rent is much lower. And the percentage of rent price less than \$2000 per much is higher. But do not rush to conclusions, let us break down the dataset further more.

Exercise 3

Use `tapply` to get following statistics for each city.

(a) The percentage of one bedrooms;

```
# insert code here save the precentage of one bedrooms by cites as
# 'pct.1b'
```

(b) the median price of one bedrooms. (You use the subset `one.bedrooms` created above)

```
# insert code here save the median of one bedrooms by cites as
# 'med.ib'
```

There are more one-bedroom rent postings at Berkeley. The median prices of one-bedrooms are less different for Berkeley and Palo Alto. The fact that the overall median price differs may probability caused by the large proportion of small apartment postings at Berkeley. How you obtain the sample may greatly influence your results. If we look at a stratified sampling dataset, where houses/apartments with 1, 2, 3 bedrooms account for 40%, 40%, 20% of the total samples of each city.

```
prop = c(0.4, 0.4, 0.2)
samples = c()
for (city in unique(craigslist$location)){
  for (b in 1:3){
    samples = c(samples, sample(which(craigslist$brs == b & craigslist$location == city), prop[b]*60))
  }
}
craigslist.srs <- craigslist[samples, ]
```

Now we look at the median rent price by cities for the stratified sampling dataset.

```
tapply(craigslist.srs$price, craigslist.srs$location, median)
```

```
##            alameda albany / el cerrito           berkeley
##             2550.0              2297.5             2837.5
##        emeryville          menlo park      mountain view
##             3095.0              3467.5             2901.5
##            oakland           palo alto       redwood city
##             2691.5              3240.0             3475.0
##           richmond           sunnyvale
##             2888.0              2892.5
```

Below is the percentage of rent price less than $2000/month by cities for the stratified sampling dataset.

```
tapply(craigslist.srs$price, craigslist.srs$location, function(x){mean(x < 2000)})
```

```
##            alameda albany / el cerrito           berkeley
##         0.16666667          0.33333333         0.05000000
##        emeryville          menlo park      mountain view
##         0.03333333          0.01666667         0.06666667
##            oakland           palo alto       redwood city
##         0.20000000          0.05000000         0.05000000
##           richmond           sunnyvale
##         0.18333333          0.11666667
```

Now the difference of price median between Berkeley and Palo Alto reduced compared to the SRS sample. This is a case where simple random samples may be misleading. And the results from stratified samples may well depend on the how you assign the proportions to each stratum. Care must be taken before you reach conclusions.

Permutation test

Exercise 4

You will do a permutation test to compare the average one bedroom apartment rent price in You will do a permutation test to compare the average one bedroom apartment rent price in Berkeley and Palo Alto.

To repeat things in R, there is a more convenient and efficient way than using for loop. The function `replicate` is designed for evaluating an expression repeatedly. For example, to get a vector of length 1000 whose elements are generated from the sum of five normal distributed samples:

```
example <- replicate(1000, sum(rnorm(5)))
```

(a) Subset the dataset to only consider one one bedroom apartment in Berkeley and Palo Alto.

```
# insert code here save the data frame of one bedroom postings
# in Berkeley and Palo Alto as
# 'subset'
```

(b) Calculate the number of postings for Berkeley in the data frame `subset`.

```
# insert code here save the number of postings for Berkeley as
# 'no.berkeley'
```

(c) Calculate the observed statistics, i.e., the difference between the mean of Berkeley and Palo Alto one bedroom rent price.

```
# insert code here save the observed statistics as
# 'stat.obs'
```

(d) Calculate the permuted statistics, repeat for 1000 times. HINT: use `sample` function to sample from a group of observations. You can use either for loop or the `replicate` function introduced above.

```
# insert code here save the observed statistics as
# 'stat.bootstrap'
set.seed(20172828)
```

(e) Calculate the p-value.

```
# insert code here save the observed statistics as
# 'p.value'
```